

1                   SELF-TESTING AND -REPAIRING

2                   FAULT-TOLERANCE INFRASTRUCTURE FOR COMPUTER SYSTEMS

3  
4  
5  
6                   BACKGROUND

7  
8                   1.     FIELD OF THE INVENTION

9  
10                   This invention relates generally to robustness (resis-  
11                   tance to failure) in computer systems; and more particularly  
12                   to novel apparatus and methods for shielding and preserving  
13                   computer systems — which can be substantially conventional  
14                   systems — from failure.

15  
16  
17                   2.     RELATED ART

18  
19                   (a)   Earlier publications — Listed below, and wholly  
20                   incorporated by reference into the present document, are ear-  
21                   lier materials in this field that will be helpful in orienting  
22                   the reader. Cross-references to these publications, by number

1 in the following list, appear enclosed in square brackets in  
2 the present document:

3  
4 [1] Intel Corp., Intel's Quality System Databook (January  
5 1998), Order No. 210997-007.

6  
7 [2] A. Avižienis and Y. He, "Microprocessor entomology: A  
8 taxonomy of design faults in COTS microprocessors", in J.  
9 Rushby and C. B. Weinstock, editors, Dependable Computing for  
10 Critical Applications 7, IEEE Computer Society Press (1999).

11  
12 [3] A. Avižienis and J. P. J. Kelly, "Fault tolerance by  
13 design diversity: concepts and experiments", Computer,  
14 17(8):67-80 (August 1984).

15  
16 [4] A. Avižienis, "The N-version approach to fault-tolerant  
17 software", IEEE Trans. Software Eng., SE11(12):1491-1501 (De-  
18 cember 1985).

19  
20 [5] M. K. Joseph and A. Avižienis, "Software fault tolerance  
21 and computer security: A shared problem", in Proc. of the

1 Annual National Joint Conference and Tutorial on Software  
2 Quality and Reliability, pages 428-36 (March 1989).  
3

4 [6] Y. He, An Investigation of Commercial Off-the-Shelf (COTS)  
5 Based Fault Tolerance, PhD thesis, Computer Science Depart-  
6 ment, University of California, Los Angeles (September 1999).  
7

8 [7] Y. He and A. Avižienis, "Assessment of the applicability  
9 of COTS microprocessors in high-confidence computing systems:  
10 A case study", in Proceedings of ICDSN 2000 (June 2000).  
11

12 [8] Intel Corp., The Pentium II Xeon Processor Server Platform  
13 System Management Guide (June 1998), Order No. 243835-001.  
14

15 [9] A. Avižienis, G. C. Gilley, F. P. Mathur, D. A. Rennels,  
16 J. A. Rohr, and D. K. Rubin. "The STAR (Self-Testing-and-Re-  
17 pairing) computer: An investigation of the theory and prac-  
18 tice of fault-tolerant computer design", IEEE Trans. Comp.,  
19 C-20(11):1312-21 (November 1971).  
20

21 [10] T. B. Smith, "Fault-tolerant clocking system", in Digest  
22 of FTCS-11, pages 262-64 (June 1981).

1 [11] Intel Corp., P6 Family Of Processors Hardware Developer's  
2 Manual (September 1998), Order No. 244001-001.

3  
4 [12] A. Avižienis, "Toward systematic design of fault-tolerant  
5 systems", Computer, 30(4):51-58 (April 1997).

6  
7 [13] "Special report: Sending astronauts to Mars", Scientific  
8 American, 282(3):40-63 (March 2000).

9  
10 [14] NASA, "Conference on enabling technology and required  
11 scientific developments for interstellar missions", OSS Ad-  
12 vanced Concepts Newsletter, page 3 (March 1999).

13  
14  
15 (b) Failure of computer systems — The purpose of a  
16 computer system is to deliver information processing services  
17 according to a specification. Such a system is said to "fail"  
18 when the service that it delivers stops or when it becomes  
19 incorrect, that is, it deviates from the specified service.

20 There are five major causes of system failure ("F"):  
21

1 (F1) permanent physical failures (changes) of its hardware  
2 components [1];  
3

4 (F2) interference with the operation of the system by external  
5 environmental factors, such as cosmic rays, electromag-  
6 netic radiation, excessive temperature, etc.;  
7

8 (F3) previously undetected design faults (also called "bugs",  
9 "errata", etc.) in the hardware and software components  
10 of a computer system that manifest themselves during  
11 operation [2-4];  
12

13 (F4) malicious actions by humans that cause the cessation or  
14 alteration of correct service: the introduction of  
15 computer "viruses", "worms", and other kinds of software  
16 that maliciously affects system operation [5]; and  
17

18 (F5) unintentional mistakes by human operators or maintenance  
19 personnel that lead to the loss or undesirable changes of  
20 system service.  
21

1 Commercial-off-the-shelf ("COTS") hardware components  
2 (memories, microprocessors, etc.) for computer systems have a  
3 low probability of failure due to failure mode F1 above [1].  
4 They contain, however, very limited protection, or none at  
5 all, against causes F2 through F5 listed above [6, 7].

6 Accordingly the related art remains subject to major  
7 problems, and the efforts outlined in the cited publications  
8 — though praiseworthy — have left room for considerable  
9 refinement.

10  
11  
12  
13 SUMMARY OF THE DISCLOSURE  
14

15 The present invention introduces such refinement. In its  
16 preferred embodiments, the present invention has several as-  
17 pects or facets that can be used independently, although they  
18 are preferably employed together to optimize their benefits.

19  
20 In preferred embodiments of its first major independent  
21 facet or aspect, the invention is apparatus for deterring fai-  
22 lure of a computing system. (The term "detering" implies

1 that the computing system is rendered less probable to fail,  
2 but there is no absolute prevention or guarantee.) The appa-  
3 ratus includes an exclusively hardware network of components,  
4 having substantially no software.

5 The apparatus also includes terminals of the network for  
6 connection to the system. In certain of the appended claims,  
7 this relationship is described as "connection to such system".

8 (In the accompanying claims generally the term "such" is  
9 used, instead of "said" or "the", in the bodies of the claims,  
10 when reciting elements of the claimed invention, for referring  
11 back to features which are introduced in preamble as part of  
12 the context or environment of the claimed invention. The pur-  
13 pose of this convention is to aid in more distinctly and em-  
14 phatically pointing out which features are elements of the  
15 claimed invention, and which are parts of its context — and  
16 thereby to more particularly claim the invention.)

17 The apparatus includes fabrication-preprogrammed hardware  
18 circuits of the network for guarding the system from failure.  
19 For purposes of this document, the term "fabrication-prepro-  
20 grammed hardware circuit" means an application-specific in-  
21 tegrated circuit (ASIC) or equivalent.

1        This terminology accordingly encompasses two main types  
2 of hardware:

3  
4    (1) a classical ASIC — i. e. a unitary, special-purpose  
5 processor circuit, sometimes called a "sequencer", fabri-  
6 cated in such a way that it substantially can perform  
7 only one program (though the program can be extremely  
8 complex, with many conditional branches and loops etc.);  
9 and

10  
11    (2) a general-purpose processor interlinked with a true read-  
12 only memory (ROM) — "true read-only" in the sense that  
13 the memory circuit and its contents substantially cannot  
14 be changed without destroying it — the memory circuit  
15 being fabricated in such a way that it contains only one  
16 program (again, potentially quite complicated), which the  
17 processor performs.

18  
19        Ordinarily either of these device types when powered up  
20 starts to execute its program — which in essence is unalter-  
21 ably preprogrammed into the device at the time of manufacture.  
22    The program in the second type of device configuration identi-



1   fied above, in which the processor reads out the program from  
2   an identifiably separate memory, is sometimes termed "firm-  
3   ware"; however, when a true ROM is used, the distinction be-  
4   tween firmware and ASIC is strongly blurred.

5         The term "fabrication-preprogrammed hardware circuit" al-  
6   so encompasses all other kinds of circuits (including optical)  
7   that follow a program which is substantially permanently manu-  
8   factured in. In particular this nomenclature explicitly en-  
9   compasses any device so described, whether or not in existence  
10  at the time of this writing.

11  
12         The foregoing may represent a description or definition  
13   of the first aspect or facet of the invention in its broadest  
14   or most general form. Even as couched in these broad terms,  
15   however, it can be seen that this facet of the invention  
16   importantly advances the art.

17         In particular, through use of a protective system that is  
18   itself all hardware the probability of failure by previously  
19   mentioned failure(F1), (F2), (F4) and (F5) in the protective  
20   system itself is very greatly reduced. Furthermore the proba-  
21   bility of failure by cause (F3) is rendered controllable by  
22   use of extremely simple hardware designs that can be qualified

1 quite completely. While these considerations alone cannot  
2 eliminate the possibility of failure in the guarded computing  
3 system, they represent an extremely important advance in that  
4 at least the protective system itself is very likely to be  
5 available to continue its protective efforts.

6  
7 Although the first major aspect of the invention thus  
8 significantly advances the art, nevertheless to optimize  
9 enjoyment of its benefits preferably the invention is prac-  
10 ticed in conjunction with certain additional features or  
11 characteristics. In particular, if the computing system is  
12 substantially exclusively made up of substantially commercial,  
13 off-the-shelf components, preferably at least one of the net-  
14 work terminals is connected to receive at least one error sig-  
15 nal generated by the computing system in event of incipient  
16 failure of that system; and at least one of the network termi-  
17 nals is connected to provide at least one recovery signal to  
18 the system upon receipt of the error signal.

19 If that preference is observed, then a subsidiary pref-  
20 erence arises: preferably the circuits include portions that  
21 are fabrication-preprogrammed to evaluate the "at least one"  
22 error signal to establish characteristics of the at least one

1 recovery signal. In other words, these circuits select or  
2 fashion the recovery signal in view of the character of the  
3 error signal.

4  
5 For the first aspect of the invention introduced above,  
6 as noted already, the computing system as most broadly con-  
7 ceived is not a part of the invention but rather is an element  
8 of the context or environment of that invention. For a vari-  
9 ant form of the first aspect of the invention, however, the  
10 protected computing system is a part of an inventive combina-  
11 tion that includes the first aspect of the invention as broad-  
12 ly defined.

13 This dual character is common to all the other aspects  
14 discussed below, and also to the various preferences stated  
15 for those other aspects: in each case a variant form of the  
16 invention includes the guarded computing system. In addition,  
17 as also mentioned above, a particularly valuable set of pref-  
18 erences for the first aspect of the invention consists of com-  
19 binations of that aspect with all the other aspects.

20 These combinations include crosscombinations of the first  
21 aspect with each of the others in turn — but also include  
22 combinations of three aspects, four and so on. Thus the most

1 highly preferred form of the invention accordingly uses all of  
2 its inventive aspects.

3  
4  
5 In preferred embodiments of its second major independent  
6 facet or aspect, the invention is apparatus for deterring fai-  
7 lure of a computing system. The apparatus includes a network  
8 of components having terminals for connection to the system,  
9 and circuits of the network for operating programs to guard  
10 the system from failure.

11 The circuits in preferred embodiments of the second facet  
12 of the invention also include portions for identifying failure  
13 of any of the circuits and correcting for the identified fai-  
14 lure. (The "circuits" whose failure is identified and correc-  
15 ted for — in this second aspect of the invention — are the  
16 circuits of the network apparatus itself, not of the computing  
17 system.)

18 For the purposes of this document, the phrase "circuits  
19 . . . for operating programs" means either fabrication-pre-  
20 programmed hardware circuit, as described above, or a firm-  
21 ware- or even software-driven circuit, or hybrids of these  
22 types. As noted earlier, all-hardware circuitry is strongly

1 preferred for practice of the invention; however, the main as-  
2 pects other than the first one do not expressly require such  
3 construction.

4  
5 The foregoing may represent a description or definition  
6 of the second aspect or facet of the invention in its broadest  
7 or most general form. Even as couched in these broad terms,  
8 however, it can be seen that this facet of the invention  
9 importantly advances the art.

10 In particular, as in the case of the first aspect of the  
11 invention, the benefits of this second aspect reside in the  
12 relative extremely high reliability of the protective apparat-  
13 us. Whereas the first aspect focuses upon benefits derived  
14 from the structural character — as such — of that apparatus,  
15 this second aspect concentrates on benefits that flow from  
16 self-monitoring and correction on the part of that apparatus.

17  
18 Although the second major aspect of the invention thus  
19 significantly advances the art, nevertheless to optimize en-  
20 joyment of its benefits preferably the invention is practiced  
21 in conjunction with certain additional features or charac-  
22 teristics. In particular, preferably the program-operating

1 portions include a section that corrects for the identified  
2 failure by taking a failed circuit out of operation.

3 In event this basic preference is followed, a subpref-  
4 erence is that the program-operating portions include a  
5 section that substitutes and powers up a spare circuit for a  
6 circuit taken out of operation. Another basic preference is  
7 that the program-operating portions include at least three of  
8 the circuits; and that failure be identified at least in part  
9 by majority vote among the at least three circuits.

10 The earlier-noted dual character of the invention — as  
11 having a variant that includes the computing system — applies  
12 to this second aspect of the invention as well as the first,  
13 and also to all the other aspects of the invention discussed  
14 below. Also applicable to this second facet and all the  
15 others is the preferability of employing all the facets  
16 together in combination with each other.

17  
18  
19 In preferred embodiments of its third major independent  
20 facet or aspect, the invention is apparatus for deterring  
21 failure of a computing system that has at least one software  
22 subsystem for conferring resistance to failure of the system;

1 the apparatus includes a network of components having termi-  
2 nals for connection to the system; and circuits of the network  
3 for operating programs to guard the system from failure.

4 The circuits include substantially no portion that inter-  
5 feres with the failure-resistance software subsystem. The  
6 foregoing may represent a description or definition of the  
7 third aspect or facet of the invention in its broadest or most  
8 general form. Even as couched in these broad terms, however,  
9 it can be seen that this facet of the invention importantly  
10 advances the art.

11 In particular, operation of this aspect of the invention  
12 advantageously refrains from tampering with protective fea-  
13 tures built into the guarded system itself. The invention  
14 thus takes forward steps toward ever-higher reliability with-  
15 out inflicting on the protected system any backward steps that  
16 actually reduce reliability.

17 Although the third major aspect of the invention thus  
18 significantly advances the art, nevertheless to optimize en-  
19 joyment of its benefits preferably the invention is practiced  
20 in conjunction with certain additional features or charac-  
21 teristics. In particular, as before, a preferred variant of

1 the invention includes the protected computing system — here  
2 particularly including the at least one software subsystem.  
3  
4

5 In preferred embodiments of its fourth major independent  
6 facet or aspect, the invention is apparatus for deterring fai-  
7 lure of a computing system that is substantially exclusively  
8 made of substantially commercial, off-the-shelf components and  
9 that has at least one hardware subsystem for generating a re-  
10 sponse of the system to failure. The apparatus includes a  
11 network of components having terminals for connection to the  
12 system; and circuits of the network for operating programs to  
13 guard the system from failure.

14 The circuits include portions for reacting to the re-  
15 sponse of the hardware subsystem. (In the "Detailed Descrip-  
16 tion" section that follows, these portions may be identified  
17 as the so-called "M-nodes" and some instances of "D-nodes".)

18 The foregoing may represent a description or definition  
19 of the fourth aspect or facet of the invention in its broadest  
20 or most general form. Even as couched in these broad terms,  
21 however, it can be seen that this facet of the invention im-  
22 portantly advances the art.



1 In particular, this facet of the invention exploits the  
2 hardware provisions of the protected computing system — i. e.  
3 the most reliable portions of that system — to establish when  
4 the protected system is actually in need of active aid. In  
5 earlier systems the only effort to intercede in response to  
6 such need was provided from the computing system itself; and  
7 that system, in event of need, was already compromised.

8 Although the fourth major aspect of the invention thus  
9 significantly advances the art, nevertheless to optimize  
10 enjoyment of its benefits preferably the invention is prac-  
11 ticed in conjunction with certain additional features or  
12 characteristics. In particular, preferably the reacting  
13 portions include sections for evaluating the hardware-subsys-  
14 tem response to establish characteristics of at least one  
15 recovery signal. When this basic preference is observed, a  
16 subpreference is that the reacting portions include sections  
17 for applying the at least one recovery signal to the system.

18  
19  
20 In preferred embodiments of its fifth major independent  
21 facet or aspect, the invention is apparatus for deterring fai-  
22 lure of a computing system that is distinct from the apparatus

1 and that has plural generally parallel computing channels.

2 The apparatus includes a network of components having termi-  
3 nals for connection to the system; and circuits of the network  
4 for operating programs to guard the system from failure.

5 The circuits include portions for comparing computational  
6 results from the parallel channels. (In the "Detailed De-  
7 scription" section that follows, these portions may be identi-  
8 fied as the so-called "D-nodes".)

9 The foregoing may represent a description or definition  
10 of the fifth aspect or facet of the invention in its broadest  
11 or most general form. Even as couched in these broad terms,  
12 however, it can be seen that this facet of the invention im-  
13 portantly advances the art.

14 In particular, this facet of the invention takes favora-  
15 ble advantage of redundant processing within the protected  
16 computing system, actually applying a reliable, objective ex-  
17 ternal comparison of outputs from the two or more internal  
18 channels. The result is a far higher degree of confidence in  
19 the overall output.

20  
21 Although the fifth major aspect of the invention thus  
22 significantly advances the art, nevertheless to optimize en-

1 joyment of its benefits preferably the invention is practiced  
2 in conjunction with certain additional features or character-  
3 istics. In particular, preferably the parallel channels of  
4 the computing system are of diverse design or origin; when  
5 outputs from parallel processing within architecturally and  
6 even commercially diverse subsystems are objectively in agree-  
7 ment, the outputs are very reliable indeed.

8 Another basic preference is that the comparing portions  
9 include at least one section for analyzing discrepancies be-  
10 tween the results from the parallel channels. If this pref-  
11 erence is in effect, then another subsidiary preference is  
12 that the comparing portions further include at least one sec-  
13 tion for imposing corrective action on the system in view of  
14 the analyzed discrepancies. In this case a still further  
15 nested preference is that the at least one discrepancy-analyz-  
16 ing section uses a majority voting criterion for resolving  
17 discrepancies.

18 When the parallel channels of the computing system are of  
19 diverse design or origin — a preferred condition, as noted  
20 above — it is further preferable that the comparing portions  
21 include circuitry for performing an algorithm to validate a  
22 match that is inexact. This is preferable because certain

1 types of calculations performed by diverse plural systems are  
2 likely to produce slightly divergent results, even when the  
3 calculations in the plural channels are performed correctly.

4 In the case of such inexactness-permissive matching, a  
5 number of alternative preferences come into play for accommo-  
6 dating the type of calculation actually involved. One is that  
7 the algorithm-performing circuitry preferably employs a degree  
8 of inexactness suited to a type of computation under compari-  
9 son; an alternative is that the algorithm-performing circuitry  
10 performs an algorithm which selects a degree of inexactness  
11 based on type of computation under comparison.

12  
13  
14 In preferred embodiments of its sixth major independent  
15 facet or aspect, the invention is apparatus for deterring fai-  
16 lure of a computing system that has plural processors; the ap-  
17 paratus includes a network of components having terminals for  
18 connection to the system; and circuits of the network for op-  
19 erating programs to guard the system from failure.

20 The circuits include portions for identifying failure of  
21 any of the processors and correcting for identified failure.

22 (In the "Detailed Description" section that follows, these

1 portions may be identified as the so-called "M-nodes" and some  
2 instances of "D-nodes".)

3 The foregoing may represent a description or definition  
4 of the sixth aspect or facet of the invention in its broadest  
5 or most general form. Even as couched in these broad terms,  
6 however, it can be seen that this facet of the invention im-  
7 portantly advances the art.

8 In particular, whereas the fifth aspect of the invention  
9 advantageously addresses the functional results of parallel  
10 processing in the protected system, this sixth facet of the  
11 invention focuses upon the hardware integrity of the parallel  
12 processors. This focus is in terms of each processor indi-  
13 vidually, as distinguished from the several processors consid-  
14 ered in the aggregate, and thus beneficially goes to a level  
15 of verification not heretofore found in the art.

16 Although the sixth major aspect of the invention thus  
17 significantly advances the art, nevertheless to optimize en-  
18 joyment of its benefits preferably the invention is practiced  
19 in conjunction with certain additional features or character-  
20 istics. In particular, preferably the identifying portions  
21 include a section that corrects for the identified failure by  
22 taking a failed processor out of operation.

1       When this basic preference is actualized, then a subpref-  
2       erence is applicable: preferably the section includes parts  
3       for taking a processor out of operation only in case of sig-  
4       nals indicating that the processor has failed permanently.  
5       Another basic preference is that the identifying portions in-  
6       clude a section that substitutes and powers up a spare circuit  
7       for a processor taken out of operation.

8  
9  
10       In preferred embodiments of its seventh major independent  
11       facet or aspect, the invention is apparatus for deterring fai-  
12       lure of a computing system. The apparatus includes a network  
13       of components having terminals for connection to the system;  
14       and circuits of the network for operating programs to guard  
15       the system from failure.

16       The circuits include modules for collecting and respond-  
17       ing to data received from at least one of the terminals. The  
18       modules include at least three data-collecting and -responding  
19       modules, and also processing sections for conferring among the  
20       modules to determine whether any of the modules has failed.

21       The foregoing may represent a description or definition  
22       of the seventh aspect or facet of the invention in its broad-

1 est or most general form. Even as couched in these broad  
2 terms, however, it can be seen that this facet of the inven-  
3 tion importantly advances the art.

4 In particular, whereas the earlier-discussed fifth aspect  
5 of the invention enhances reliability through comparison of  
6 processing results among subsystems within the protected com-  
7 puting system, this seventh facet of the invention looks to  
8 comparison of modules in the protective apparatus itself — to  
9 attain an analogous upward step in reliability of the hybrid  
10 overall system.

11 Although the seventh major aspect of the invention thus  
12 significantly advances the art, nevertheless to optimize en-  
13 joyment of its benefits preferably the invention is practiced  
14 in conjunction with certain additional features or charac-  
15 teristics. In particular, these preferences as mentioned ear-  
16 lier include crosscombinations of the several facets or as-  
17 pects, and also the dual character of the invention — i. e.,  
18 encompassing a variant overall combination which includes the  
19 protected computing system.

1 In preferred embodiments of its eighth major independent  
2 facet or aspect, the invention is apparatus for deterring fai-  
3 lure of a computing system. The latter system is substantial-  
4 ly exclusively made of substantially commercial, off-the-shelf  
5 components, and has at least one subsystem for generating a  
6 response of the system to failure — and also has at least one  
7 subsystem for receiving recovery commands.

8 The apparatus includes a network of components having  
9 terminals for connection to the system between the response-  
10 generating subsystem and the recovery-command-receiving sub-  
11 system. It also has circuits of the network for operating  
12 programs to guard the system from failure.

13 The circuits include portions for interposing analysis  
14 and a corrective reaction between the response-generating sub-  
15 system and the command-receiving subsystem. The foregoing may  
16 represent a description or definition of the eighth aspect or  
17 facet of the invention in its broadest or most general form.  
18 Even as couched in these broad terms, however, it can be seen  
19 that this facet of the invention importantly advances the art.

20 In particular, earlier fault-detecting efforts have con-  
21 centrated upon feeding back corrective reaction within the  
22 protected system itself. Such prior attempts are flawed in



1 that generally commercial, off-the-shelf systems intrinsically  
2 lack both the reliability and the analytical capability to po-  
3 lice their own failure modes.

4 Although the eighth major aspect of the invention thus  
5 significantly advances the art, nevertheless to optimize en-  
6 joyment of its benefits preferably the invention is practiced  
7 in conjunction with certain additional features or character-  
8 istics. In particular, preferably the general preferences  
9 mentioned above (e. g. as to the seventh facet) are equally  
10 applicable here.

11  
12  
13 All of the foregoing operational principles and advantag-  
14 es of the present invention will be more fully appreciated  
15 upon consideration of the following detailed description, with  
16 reference to the appended drawings, of which:

1 BRIEF DESCRIPTION OF THE DRAWINGS

2  
3 Fig. 1 is a partial block diagram, very schematic, of a  
4 two-ring architecture used for preferred embodiments of the  
5 invention;

6 Fig. 2 is a like view, but expanded, of the inner ring  
7 including a group of components called the "M-cluster";

8 Fig. 3 is an electrical schematic of an  $n$ -bit comparator  
9 and switch used in preferred embodiments;

10 Fig. 4 is a set of two like schematics — Fig. 4a showing  
11 one "A-node" or "A-port" (namely the "a" half of a self-check-  
12 ing A-pair "a" and "b"), and Fig. 4b showing connections of  
13 A-nodes "a" and "b" with their C-node;

14 Fig. 5 is a like schematic showing one M-node (monitor  
15 node) from a five-node M-cluster;

16 Fig. 6 is a view like Figs. 1 and 2, but showing the core  
17 of the M-cluster;

18 Fig. 7 is a schematic like Figs. 3 through 5 but showing  
19 one self-checking S3-node (b-side blocks not shown) in a total  
20 set of four S3-nodes;

21 Fig. 8 is a set of three flow diagrams — Fig. 8a showing  
22 a power-on sequence for the M-cluster, controlled by S3-nodes,

1 Fig. 8 b showing a power-on sequence for the outer ring (one  
2 node), controlled by an M-cluster, and Fig. 8c showing a pow-  
3 er-off sequence for the invention;

4 Fig. 9 is a schematic like Figs. 3 through 5, and 7, but  
5 showing one of a self-checking pair of D-nodes, namely node  
6 "a" (the identical twin D-node "b" not shown); and

7 Fig. 10 is a block diagram, highly schematic, of a fault-  
8 tolerant chain of interstellar spacecraft embodying certain  
9 features of the invention.

10  
11 A key to symbols and callouts used in the drawings ap-  
12 pears at the end of this text, preceding the claims.

13  
14  
15  
16 DETAILED DESCRIPTION

17 OF THE PREFERRED EMBODIMENTS

18  
19 1. SYSTEM ELEMENTS

20  
21 Preferred embodiments of the present invention provide a  
22 so-called "fault-tolerance infrastructure" (FTI) that is a

1 system composed of four types of special-purpose controllers  
2 which will be called "nodes". The nodes are ASICs (applica-  
3 tion-specific integrated circuits) that are controlled by  
4 hardwired sequencers or by microcode.

5 The preferred embodiments employ no software. The four  
6 kinds of nodes will be called:

- 7
- 8 (1) A-nodes (adapter nodes);
  - 9 (2) M-nodes (monitor nodes);
  - 10 (3) D-nodes (decision nodes); and
  - 11 (4) S3-nodes (startup, shutdown, and survival nodes).
- 12

13 The purpose of the FTI is to provide protection against  
14 all five causes of system failure for a computing system that  
15 can be substantially conventional and composed of COTS compo-  
16 nents, called C-nodes (computing nodes). Merely for the sake  
17 of simplicity — and tutorial clarity in emphasizing the capa-  
18 bilities of the invention — this document generally refers to  
19 the C-nodes as made up of COTS components, or as a "COTS sys-  
20 tem"; however, it is to be understood that the invention is  
21 not limited to protection of COTS systems and is equally ap-  
22 plicable to guarding custom systems.

1       The C-nodes are connected to the A-nodes and D-nodes of  
2       the FTI in the manner described subsequently. The C-nodes can  
3       be COTS microprocessors, memories, and components of the sup-  
4       porting chipset in the COTS computer system that will be  
5       called the "client system" or simply the "client".

6       The following protection for the client system is provi-  
7       ded when it is connected to the FTI.

8  
9       (1) The FTI provides error detection and recovery support  
10       when the client COTS system is affected by physical fai-  
11       lures of its components (F1) and by external interference  
12       (F2). The FTI provides power switching for unpowered  
13       spare COTS components of the client system to replace  
14       failed COTS components (F1) in long-duration missions.

15  
16       (2) The FTI provides a "shutdown-hold-restart" recovery se-  
17       quence for catastrophic events (F2, F3, F4) that affect  
18       either the client COTS system or both the COTS and FTI  
19       systems. Such events are: a "crash" of the client COTS  
20       system software, an intensive burst of radiation, tempo-  
21       rary outage of client COTS system power, etc.

1 (3) The FTI provides (by means of the D-nodes) the essential  
2 mechanisms to detect and to recover from the manifesta-  
3 tions of software and hardware design faults (F3) in the  
4 client system.

5 This is accomplished by the implementation of design  
6 diversity [3, 4]. Design diversity is the implementation  
7 of redundant channel computation (duplication with com-  
8 parison, triplication with voting, etc.) in which each  
9 channel (i. e. C-node) employs independently designed  
10 hardware and software, while the D-node serves as the  
11 comparator or voter element. Design diversity also pro-  
12 vides detection and neutralization of malicious software  
13 (F4) and of mistakes (F5) by operators or maintenance  
14 personnel [5].

15  
16 Finally, the nodes and interconnections of the FTI are  
17 designed to provide protection for the FTI system itself as  
18 follows.

19  
20 (1) Error detection and recovery algorithms are incorporated  
21 to protect against causes (F1) and (F2).  
22

1 (2) The absence of software in the FTI provides immunity  
2 against causes (F4) and (F5).  
3

4 (3) The overall FTI design allows the introduction of diverse  
5 hardware designs for the A-, M-, S3-, and D-nodes in or-  
6 der to provide protection against cause (F3), i. e. hard-  
7 ware design faults. Such protection may prove not be  
8 necessary, since low complexity of the node structure  
9 should allow complete verification of the node designs.  
10

11 When interconnected in the manner described below, the  
12 FTI and the client COTS computing system form a high-perfor-  
13 mance computing system that is protected against all five  
14 system failure causes (F1)-(F5). For purposes of the present  
15 document this system will be called a "diversifiable self-  
16 testing and -repairing system" ("DiSTARS").  
17  
18

## 19 2. ARCHITECTURE OF DiSTARS 20

21 (a) The DiSTARS Configuration — The structure of a pre-  
22 ferred embodiment of DiSTARS conceptually consists of two con-

centric rings (Fig. 1): an Outer Ring and an Inner Ring. The Outer Ring contains the client COTS system, composed of Computing Nodes or C-nodes 11 (Fig. 1) and their System Bus 12.

The C-nodes are either high-performance COTS processors (e. g. Pentium II) with associated memory, or other COTS elements from the supporting chipset (I/O controllers, etc.), and other subsystems of a server platform [8]. The Outer Ring is supplemented with custom-designed Decision Nodes or "D-nodes" 13 that communicate with the C-nodes via the System Bus 12. The D-nodes serve as comparators or voters for inputs provided by the C-nodes. They also provide the means for the C-nodes to communicate with the Inner Ring. Detailed discussion of the D-node is presented later.

The Inner Ring is a custom-designed system composed of Adapter Nodes or "A-nodes" 14 and a cluster of Monitor Nodes, or "M-nodes", called the M-cluster 15. The A-nodes and the M-nodes communicate via the Monitor Bus or "M-bus" 16. Every A-node also has a dedicated A-line 17 for one-way communication to the M-nodes. The custom-designed D-nodes 13 of the Outer Ring contain embedded A-ports 18 that serve the same purpose as the external A-nodes of the C-node processors.



1       The M-cluster serves as a fault-tolerant controller of  
2       recovery management for the C- and D-nodes in the Outer Ring.  
3       The M-cluster employs hybrid redundancy (triplication and vot-  
4       ing, with unpowered spares) to assure its own continuous  
5       availability. It is an evolved descendant of the Test-and-  
6       Repair processor of the JPL-STAR computer [9]. Two dedicated  
7       A-nodes are connected to every C-node, and every D-node con-  
8       tains two A-ports. The A-nodes and A-ports serve as the input  
9       and output devices of the M-cluster: they relay error signals  
10      and other relevant outputs of the C- and D-nodes to the M-  
11      cluster and return M-cluster responses to the appropriate C-  
12      or D-node inputs.

13      The custom-designed Inner Ring and the D-nodes provide an  
14      FTI that assures dependable operation of the client COTS com-  
15      puting system composed of the C-nodes. The infrastructure is  
16      generic; that is, it can accommodate any client system (set of  
17      Outer Ring C-node chips) by providing them with the A-nodes  
18      and storing the proper responses to A-node error messages in  
19      the M-nodes. Fault-tolerance techniques are extensively used  
20      in the design of the infrastructure's components.

21      The following discussion explains the functions and  
22      structure of the inner ring elements (Fig. 2) — particularly

1 the A- and M-nodes, the operation of the M-cluster, and the  
2 communication between the M-cluster and the A-nodes. Unless  
3 explicitly stated otherwise, the A-ports are structured and  
4 behave like the A-nodes. The D-nodes are discussed in Section  
5 3 below.

6  
7 (b) The Adapter Nodes (A-Nodes) and A-lines — The pur-  
8 pose of an A-node (Fig. 4a) is to connect a particular C-node  
9 to the M-cluster that provides Outer Ring recovery management  
10 for the client COTS system. The functions of an A-node are  
11 to:

- 12  
13 1. transmit error messages that are originated by its C-node  
14 to the M-cluster;
- 15  
16 2. transmit recovery commands from the M-cluster to its  
17 C-node;
- 18  
19 3. control the power switch of the C-node and its own fuse  
20 according to commands received from the M-cluster; and  
21  
22 4. report its own status to the M-cluster.

1        Every C-node is connected to an A-pair that is composed  
2 of two A-nodes, three CS units CS1, CS2, CS3 (Fig. 4b), one OR  
3 Power Switch 415 that provides power to the C-node and one  
4 Power Fuse 416 common to both A-nodes and the CS units. The  
5 internal structure of a CS unit is shown in Fig. 3. The two  
6 A-nodes (Fig. 4a) of the A-pair have, in common, a unique  
7 identification or "ID" code 403 that is associated with their  
8 C-node; otherwise, all A-nodes are identical in their design.  
9 They encode the error signal outputs 431 of their C-node and  
10 decode the recovery commands 407 to serve as inputs 441a to  
11 the comparator CS1 that provides command inputs to the C-node.

12        As an example, consider the Pentium II processor as a  
13 C-node. It has five error signal output pins: AERR (address  
14 parity error), BINIT (bus protocol violation), BERR (bus non-  
15 protocol error), IERR (internal non-bus error), and THERMTRIP  
16 (thermal overrun error) which leads to processor shutdown. It  
17 is the function of the A-pair to communicate these signals to  
18 the M-cluster. The Pentium II also has six recovery command  
19 input pins: RESET, INIT (initialize), BINIT (bus initialize),  
20 FLUSH (cache flush), SMI (system management interrupt), and  
21 NMI (non-maskable interrupt). The A-pair can activate these  
22 inputs according to the commands received from the M-cluster.

1        Each A-node has a separate A-line 444a, 444b for messages  
2 to the M-cluster. The messages are:

- 3
- 4            (1) All is well, C-node powered,
  - 5            (2) All is well, C-node unpowered,
  - 6            (3) M-bus request,
  - 7            (4) Transmitting on M-bus, and
  - 8            (5) Internal A-node fault.
- 9

10       All A-pairs of the Inner Ring are connected to the M-bus,  
11 which provides two-way communication with the M-cluster as  
12 discussed in the next subsection.

13       The outputs 441a, 441b (Fig. 4b) of the A-pair to the  
14 C-node, outputs 442a, 442b to the C-node power switch and  
15 outputs 445a, 445b to the M-bus are compared in Comparator  
16 circuits CS1, CS2, CS3. In case of disagreement, the outputs  
17 441, 442, 445 are inhibited (assume the high-impedance third  
18 state Z) and an "Internal fault" message is sent on the two  
19 A-lines 444a, 444b (Fig. 4a). The single exception is the  
20 C-node Power-Off command. One Power-Off command is sufficient  
21 to turn C-node power 446 (Fig. 4b) off after the failure of  
22 one A-node in the pair.

1       The A-pair remains powered by Inner Ring power 426 when  
2       Outer Ring power 446 to its C-node is off — i. e., when the  
3       C-node is a spare or has failed. The failure of one A-node in  
4       the self-checking A-pair turns off the power of its C-node. A  
5       fuse 416 is used to remove power from a failed A-pair, thus  
6       protecting the M-bus against "babbling" outputs from the  
7       failed A-pair. Clock synchronization signals 425a (Fig. 4a)  
8       are delivered from the M-cluster. The low complexity of the  
9       A-node allows the packaging of the A-pair and power switch as  
10      one IC device.

11  
12      (c) The Monitor (M-) Nodes, M-Cluster and M-Bus — The  
13      purpose of the Monitor Node (M-node, Fig. 5) is to collect  
14      status and error messages from one or more (and in the aggre-  
15      gate all) A-nodes, to select the appropriate recovery action,  
16      and to issue recovery-implementing commands to the A-node or  
17      nodes via the Monitor Bus (M-Bus). To assure continuous  
18      availability, the M-nodes are arranged in a hybrid redundant  
19      M-cluster — with three powered M-nodes in a triplication-and-  
20      voting mode, or as it is often called "triple modular redun-  
21      dancy" (TMR); and also with unpowered spare M-nodes. The vot-  
22      ing on output commands takes place in Voter logic 410 (Fig.

1 4a) located in the A-nodes. A built-in self-test (BIST) se-  
2 quence 408 is provided in every M-node.

3 The M-bus is controlled by the M-cluster and connected to  
4 all A-nodes, as discussed in the previous section. All messa-  
5 ges are error-coded, and spare bus lines are provided to make  
6 the M-bus fault-tolerant. Two kinds of messages are sent to  
7 the A-pairs by the M-cluster: (1) an acknowledgment of A-pair  
8 request (on their A-lines 444a, 444b) that allocates a time  
9 slot on the M-bus for the A-pair error message; and (2) a com-  
10 mand in response to the error message.

11 An M-node stores two kinds of information: static (per-  
12 manent) and dynamic. The static (ROM) data 505 (Fig. 5) con-  
13 sist of:

14  
15 (1) predetermined recovery command responses to A-pair error  
16 messages,

17  
18 (2) sequences for M-node recovery and replacement in the  
19 hybrid-redundant M-cluster, and

20  
21 (3) recovery sequences for catastrophic events — discussed  
22 in subsection 2(f).

1 The dynamic data consist of:

- 2
- 3 (1) Outer Ring configuration status 504 (active, spare,  
4 failed node list),  
5
- 6 (2) Inner Ring configuration status 503 and system time 502,  
7
- 8 (3) a "scratchpad" store 501, 506, 507, 509, 510 for current  
9 activity: error messages still active, requests waiting,  
10 etc., and  
11
- 12 (4) an Inner Ring activity log (also in 506).  
13

14 The configuration status and system time are the critical data  
15 that are also stored in nonvolatile storage in the S3 nodes of  
16 the Cluster Core — discussed in subsection 2(d).

17 As long as all A-nodes continue sending "All is well"  
18 messages on their A-lines (525 through 528 and so on), the  
19 M-cluster issues 541 "All is well" acknowledgments. When an  
20 "M-bus request" message arrives on two A-lines that come from  
21 a single A-pair that has a unique C-node ID code, the M-clus-  
22 ter sends 541 (on the M-bus) the C-node ID followed by the

1 "Transmit" command. In response, the A-pair sends 522 (on the  
2 M-bus) its C-node ID followed by an Error code originated by  
3 the C-node. The M-nodes return 541 the C-node ID followed by  
4 a Recovery command for the C-node. The A-pair transmits the  
5 command to the C-node and returns 522 an acknowledgment: its  
6 C-node ID followed by the command it forwarded to the C-node.  
7 At the times when an A-pair sends a message on the M-bus, its  
8 A-lines send the "Transmitting" status report. This feature  
9 allows the M-cluster to detect cases in which a wrong A-pair  
10 responds on the M-bus. The A-pair also sends an Error message  
11 on that bus if its voters detect disagreements between the  
12 three M-cluster messages received on the M-bus.

13 When the A-pair comparators CS1, CS2, CS3 (Fig. 3b) de-  
14 tect a disagreement, the A-lines send an "Internal Fault" mes-  
15 sage to the M-cluster, which responds (on the M-bus) with the  
16 C-node ID followed by the "Reset A-pair" command. Both of the  
17 A-nodes of the A-pair attempt to reset to an initial state,  
18 but do not change the setting of the C-node power switch.  
19 Success causes "All is well" to be sent on the A-lines to the  
20 M-cluster. In case of failure to reset, the A-lines continue  
21 sending the "Internal Fault" message.



1       The M-cluster sends "Power On" and "Power Off" commands  
2       522 (Fig. 5) as part of a replacement or reconfiguration se-  
3       quence for the C-nodes. They are acknowledged immediately but  
4       power switching itself takes a relatively long time. When  
5       switching is completed, the A-pair issues an "M-bus Request"  
6       on its A-lines and then reports 522 on the M-bus the success  
7       (or failure) of the switching to the M-cluster via the M-bus.

8       When the M-cluster determines that one A-node of an  
9       A-pair has permanently failed, it sends an "A-pair Power Off"  
10      message 541 to that A-pair. The good A-node receives the mes-  
11      sage, turns C-node power 446 (Fig. 4b) off — if it was on —  
12      and then permanently opens (by 443a or 443b) the A-pair power  
13      fuse 416. The M-cluster receives confirmation via the A-lines  
14      444a, 444b, (Fig. 4a) which assume the "no power" state. This  
15      irreversible command is also used when a C-node fails perma-  
16      nently and must be removed from the Outer Ring.

17  
18      (d) The M-Cluster Core — The Core (Fig. 6) of the ear-  
19      lier-introduced M-cluster (Fig. 2) includes a set of S3-nodes  
20      (Fig. 7) and communication links. As mentioned earlier, "S3"  
21      stands for Startup, Shutdown, Survival). The M-nodes (Fig. 5)  
22      have dedicated "Disagree" 545, "Internal Error" 544 and "Re-

1 placement Request" 543 outputs to all other M-nodes and to the  
2 S3-nodes. The IntraCluster-Bus or IC-Bus 602 (Fig. 6) inter-  
3 connects all M-nodes.

4 The purpose of the S3 nodes is to support the survival of  
5 DiSTARS during catastrophic events, such as intensive bursts  
6 of radiation or temporary loss of power. Every S3-node is a  
7 self-checking pair with its own backup (battery) power 707  
8 (Fig. 7). At least two S3 nodes are needed to attain fault-  
9 tolerance, and the actual number needed depends on the mission  
10 length without external repair.

11 The functions of the S3 nodes are to:

- 12  
13 (1) execute the "power-on" and "power-off" sequences (Fig. 8)  
14 for DiSTARS;
- 15  
16 (2) provide fault-tolerant clock signals 720 (Fig. 7);
- 17  
18 (3) keep System Time 702a and System Configuration 704a, 705a  
19 data in nonvolatile, radiation-hardened registers; and  
20

1 (4) control M-node power switches 511 (Fig. 5), and I-Ring  
2 power 450 (Fig. 4b) to the A-pairs, in order to support  
3 M-cluster recovery.  
4

5 More details of S3-node operation follow in subsection 2(f).

6 Each self-checking S3 node has its own clock generator  
7 701 (Fig. 7). The hardware-based fault-tolerant clocking  
8 system developed at the C. S. Draper Laboratory [10] is the  
9 most suitable for the M-cluster.  
10

11 (e) Error Detection and Recovery in the M-cluster — At  
12 the outset, the three powered M-nodes 201a, 201b, 201c (Fig.  
13 2) are in agreement and contain the same dynamic data. They  
14 operate in the triple modular redundancy (TMR) mode. Three  
15 commands are issued in sequence on the M-bus 202 and voted  
16 upon in the A-nodes 410 (Fig. 4a). During operation of the  
17 M-cluster, one M-node may issue an output different from the  
18 other two, or one M-node may detect an error internally and  
19 send an "Internal Error" signal on a dedicated line 544 (Fig.  
20 5) to the other M-nodes. The cause may be either a "soft"  
21 error due to a transient fault, or a "hard" error due to phys-  
22 ical failure.

1 M-node output disagreement detection in the TMR mode  
2 (when one M-node is affected by a fault) works as follows.  
3 The three M-nodes 201a, 201b, 201c (Fig. 2) place their out-  
4 puts on the M-bus 202 in a fixed sequence. Each M-node com-  
5 pares its output to the outputs of the other two nodes, re-  
6 cords one or two disagreements, and sends one or two "Disa-  
7 gree" messages to the other M-nodes on a dedicated line 545  
8 (Fig. 5). The affected M-node will disagree twice, while the  
9 good M-nodes will disagree once each and at the same time,  
10 which is the time slot of the affected M-node.

11 Following error detection, the following recovery se-  
12 quence is carried out by the two good M-nodes.

- 13
- 14 (1) Identify the affected M-node or the M-node that sent the  
15 Internal Error message, and enter the Duplex Mode of the  
16 M-cluster.
  - 17
  - 18 (2) Attempt "soft" error recovery by reloading the dynamic  
19 data of the affected M-node from the other two M-nodes  
20 and resume TMR operation.
  - 21

1 (3) If Step (2) does not lead to agreement, send request for  
2 replacement 543 (Fig. 5) of the affected M-node to the  
3 S3-nodes.

4  
5 (4) The S3-nodes replace the affected M-node and send "Resume  
6 TMR" command 726 (Fig. 7) .

7  
8 (5) Load the new M-node with dynamic data from the other two  
9 M-nodes and resume TMR operation.

10  
11 During the recovery sequence, the two good (agreeing)  
12 M-nodes 601a, 601b (Fig. 6) operate in the Duplex Mode, in  
13 which they continue to communicate with the A-nodes and con-  
14 currently execute the recovery steps (2) through (5). The  
15 Duplex Mode becomes the permanent mode of operation if only  
16 two good M-nodes are left in the M-cluster. Details of the  
17 foregoing M-cluster recovery sequence are discussed next.

18  
19 Step (1): Entering Duplex Mode. The simultaneous disagree-  
20 ment 527 (Fig. 5) by the good M-nodes 601a, 601b (Fig. 6)  
21 during error detection causes the affected M-node c1 to enter  
22 the "Hold" mode, in which it inhibits its output 541 (Fig. 5)

1 to the M-bus and does not respond to inputs on the A-lines.  
2 It also clears its "Disagree" output 645. If the affected  
3 node 601c (Fig. 6) does not enter the "Hold" mode, step (3) is  
4 executed to cause its replacement. An M-node similarly enters  
5 the "Hold" mode when it issues an Internal Error message 544  
6 (Fig. 5) to the other two M-nodes, which enter the Duplex Mode  
7 at that time. It may occur that all three M-nodes disagree,  
8 i. e., each one issues two "Disagree" signals 545, or that two  
9 or all three M-nodes signal Internal Error 544. These cata-  
10 strophic events are discussed in subsection 2(f).

11 The two good M-nodes 601a, 601b (Fig. 6) still send three  
12 commands to the A-nodes in Duplex Mode during steps (2)-(5).  
13 During t1 and t2 they send their outputs to the M-bus and  
14 compare. An agreement causes the same command to be sent  
15 during t3; disagreement invokes a retry, then catastrophic  
16 event recovery. The good M-nodes continue operating in Duplex  
17 Mode if a spare M-node is not available after the affected  
18 node has been powered off in step (3). TMR operation is  
19 permanently degraded to Duplex in the M-cluster.

20  
21 Step (2): Reload Dynamic Data of the Affected M-node (assum-  
22 ing M-node 601c [Fig. 6] is affected). An IntraCluster Bus or

1 IC-bus 2 is used for this purpose. At times t1 and t2 the  
2 good M-nodes 601a, 601b place the corresponding dynamic data  
3 on the IC-Bus 602; at time t3 the affected node 601c compares  
4 and stores it. The good nodes also compare their outputs.  
5 Any disagreement causes a repetition of times t1, t2, t3. A  
6 further disagreement between good nodes is a catastrophic  
7 event. After reloading is completed, it is validated: the  
8 affected node reads out its data, and the good nodes compare  
9 it to their copies. A disagreement leads to step (3), i. e.  
10 power-off for the affected node; otherwise the M-cluster  
11 returns to TMR operation.

12  
13 Steps (3) and (4): Power Switching. Power switching 511  
14 (Fig. 5) is a mechanism for removing failed M-nodes and bring-  
15 ing in spares in the M-cluster. Failed nodes with power on  
16 can lethally interfere with M-cluster functioning; therefore  
17 very dependable switching is essential. The power-switching  
18 function 730 (Fig. 7) is performed by the S3-nodes in the  
19 Cluster Core. They maintain a record of M-cluster status in  
20 nonvolatile storage 705a. Power is turned off for the failed  
21 M-node, the next spare is powered up, BIST is executed, and  
22 the "Resume TMR" command 530 (Fig. 5) is sent to the M-nodes.

1 Step (5): Loading a New M-node. When the "Resume TMR" com-  
2 mand of step (4) is received, the new M-node must receive the  
3 dynamic data from the two good M-nodes. The procedure is the  
4 same as step (2).

5  
6  
7 (f) Recovery after Catastrophic Events — Up to this  
8 point recovery has been defined in response to an error signal  
9 from one C-node, A-node, or M-node for which the M-cluster had  
10 a predetermined recovery command or sequence. These recover-  
11 ies are classified as local and involve only one node.

12 It is possible, however, for error signals to originate  
13 from two or more nodes concurrently (or close in time). A few  
14 such cases have been identified as "catastrophic" events  
15 (c-events) in the preceding discussion. It is not practical  
16 to predetermine unique recovery for each c-event; therefore,  
17 more general catastrophe-recovery (c-recovery) procedures must  
18 be devised.

19 In general, I can distinguish c-events that affect the  
20 Outer Ring only, and c-events that affect the Inner Ring as  
21 well. For the Outer Ring a c-event is a crash of system soft-  
22 ware that requires a restart with Inner Ring assistance. The



1 Inner Ring does not employ software, thus assuming well proven  
2 ASIC programming its crash cannot occur in the absence of  
3 hardware failure (F1), (F2).

4 There are, however, adverse physical events of the (F1)  
5 and (F2) types that can cause c-events for the entire DiSTARS.  
6 Examples are: (1) external interference by radiation; (2)  
7 fluctuations of ambient temperature; (3) temporary instability  
8 or outage of power; (4) physical damage to system hardware.

9 The predictable manifestations of these events in DiSTARS  
10 are: (1) halt in operation due to power loss; (2) permanent  
11 failures of system components (nodes) and/or communication  
12 links; (3) crashes of Outer Ring application and system soft-  
13 ware; (4) errors in or loss of M-node data stored in volatile  
14 storage; (5) numerous error messages from the A-nodes that ex-  
15 ceed the ability of M-cluster to respond in time; (6) double  
16 or triple disagreements or Internal Error signals in the  
17 M-cluster TMR or Duplex Modes.

18 The DiSTARS embodiments now most highly preferred employ  
19 a System Reset procedure in which the S3-nodes execute a "pow-  
20 er-off" sequence (Fig. 8c) for DiSTARS on receiving a c-event  
21 signal either from sensors (radiation level, power stability,  
22 etc.) or from the M-nodes. System Time 702a (Fig. 7) and

1 DiSTARS configuration data 704a, 705a are preserved in the ra-  
2 diation-hardened, battery-powered S3-nodes. The "power-on"  
3 sequence (Figs. 8a, 8b) is executed when the sensors indicate  
4 a return to normal conditions.

5 Outer Ring power is turned off when the S3-node sends the  
6 signal 729 (Fig. 7) to remove power from the A-pairs, thus  
7 setting all C-node switches to the "Off" position. M-node  
8 power is directly controlled by the S3-node output 730.

9 The "power-on" sequence for M-nodes (Fig. 8a) begins with  
10 the S3-nodes applying power and executing BIST to find three  
11 or two good M-nodes, loading them via the IC-Bus with critical  
12 data, then applying I-Ring power to the A-pairs. The sequence  
13 continues with sending the "Outer Ring Power On" command 727  
14 (Fig. 7) to the M-cluster.

15 To start the "power on" sequence for C- and D-nodes (Fig.  
16 8b) the M-cluster commands (on the M-bus) "Power-On" followed  
17 by BIST sequentially for the C-nodes and D-nodes of the Outer  
18 Ring, and the system returns to an operating condition,  
19 possibly having lost some nodes due to the catastrophic event.

20 Currently preferred embodiments are equipped with only  
21 the "power-off" sequence to respond to c-events. The inven-  
22 tion, however, contemplates introducing less drastic and fas-

1 ter recovery sequences for some less harmful c-events. Ex-  
2 periments in progress with the prototype DiSTARS system ad-  
3 dress development of such sequences.

### 6 3. THE DECISION (D-) NODES AND DIVERSIFICATION

8 (a) The rationale for D-Nodes — The A-nodes in the dis-  
9 cussion thus far have been the only means of communication  
10 between the Inner and Outer Rings, and they convey only very  
11 specific C-node information. A more-general communication  
12 link is needed. The Outer Ring may need configuration data  
13 and activity logs from the M-cluster, or to command the pow-  
14 ering up or down of some C-nodes for power management reasons.  
15 An InterRing communication node beneficially acts as a link  
16 between the System Bus of the Outer Ring and the M-bus of the  
17 Inner Ring.

18 A second need of the Outer Ring is enhanced error detec-  
19 tion coverage. For example, as described in subsection 2(b),  
20 the Pentium II has only five error-signal outputs of very  
21 general nature, and in a recent study [6, 7] their coverage  
22 was estimated to be very limited. The original design of the

1 P6 family of Intel processors included the FRC (functional  
2 redundancy checking) mode of operation in which two processors  
3 could be operated in the Master/Checker mode, providing very  
4 good error confinement and high error detection coverage. De-  
5 tection of an error was indicated by the FRCERR signal. Quite  
6 surprisingly and without explanation, the FRCERR pin was re-  
7 moved from the specification in April 1998, thus effectively  
8 canceling the use of the FRC mode long after the P6 processors  
9 reached the market.

10 In fairness it should be noted that other processor ma-  
11 kers have never even tried to provide Master/Checker duplexing  
12 for their high-performance processors with low error detection  
13 coverage. An exception is the design of the IBM G5 and G6  
14 processors [7].

15 This observation explains the inclusion of a custom Deci-  
16 sion Node (D-node) on the Outer Ring System Bus that can serve  
17 as an external comparator or voter for the C-node COTS proces-  
18 sors. It is even more important that the D-node also be able  
19 to support design diversity by providing the appropriate de-  
20 cision algorithms for N-version programming [4] employing di-  
21 verse processors as the C-nodes of the Outer Ring.

1       The use of processor diversity has become important for  
2       dependable computing because contemporary high-performance  
3       processors contain significant numbers of design faults. For  
4       example, a recent study shows that in the Intel P6 family  
5       processors from forty-five to 101 design faults ("errata")  
6       were discovered (as of April 1999) after design was complete,  
7       and that from thirty to sixty of these design faults remain in  
8       the latest versions ("steppings") of these processors [2].

9  
10       (b) Decision Node (D-Node) Structure and Functions —

11       The D-nodes (Fig. 9) need to be compatible with the C-nodes on  
12       the System Bus and also embed Adapter (A-) Ports analogous to  
13       the A-nodes that are attached to C-nodes. The functions of  
14       the D-nodes are:

- 15  
16       (1) to transmit messages originated by C-node software to the  
17       M-cluster;  
18  
19       (2) to transfer M-cluster data to the C-nodes that request  
20       it;  
21

- 1 (3) to accept C-node outputs for comparison or voting and to  
2 return the results to the C-nodes;  
3  
4 (4) to provide a set of decision algorithms for N-version  
5 software executing on diverse processors (C-nodes), to  
6 accept cross-check point outputs and return the results;  
7  
8 (5) to log disagreement data on the decisions; and  
9  
10 (6) to provide high coverage and fault tolerance for the  
11 execution of the above functions.

12  
13 Ideally the programs of the C-nodes are written with pro-  
14 visions to take advantage of D-node services. The relatively  
15 simple functions of the D-node can be implemented by microcode  
16 and the D-node response can be very fast. Another advantage  
17 of using the D-node for decisions (as opposed to doing them in  
18 the C-nodes) is the high coverage and fault tolerance of the  
19 D-node (implemented as a self-checking pair) that assures er-  
20 ror-free results.

21 The Adapter Ports (A-Ports) of the D-node need to provide  
22 the same services that the A-nodes provide to the C-nodes,

1 including power switching for spare D-node utilization. In  
2 addition, the A-ports must also serve to relay appropriately  
3 formatted C-node messages to the M-cluster, then accept and  
4 vote on M-cluster responses. The messages are requests for  
5 C-node power switching, Inner and Outer Ring configuration  
6 information, and M-cluster activity logs. The D-node can  
7 periodically request and store the activity logs, thus reduc-  
8 ing the amount of dynamic storage in the M-nodes. The D-nodes  
9 can also serve as the repositories of other data that may  
10 support M-cluster operations, such as the logs of disagree-  
11 ments during D-node decisions, etc.

12 The relatively simple D-nodes can effectively compensate  
13 for the low coverage and poor error containment of contempo-  
14 rary processors (e. g. Pentium II) by allowing their duplex or  
15 TMR operation with reliable comparisons or voting and with  
16 diverse processors executing N-version software for the tol-  
17 erance of software and hardware design faults.

#### 4. A PROOF-OF-CONCEPT EXPERIMENTAL SYSTEM

The Two Ring configuration, with the Inner Ring and the D-nodes providing the fault-tolerance infrastructure for the Outer Ring of C-nodes that is a high-performance "client" COTS computer, is well defined and complete.

Many design choices and tradeoffs, however, remain to be evaluated and chosen. A prototype DiSTARS system for experimental evaluation uses a four-processor symmetric multiprocessor configuration [11] of Pentium II processors with the supporting chipset as the Outer Ring. The Pentium II processors serve as C-nodes. The S3-nodes, M-nodes, D-nodes, A-nodes and A-ports are being implemented by Field-Programmable Gate Arrays (FPGAs).

This development includes construction of power switches and programming of typical applications running on duplex C-nodes that use the D-node for comparisons; and diversification of C-nodes and N-version execution of typical applications. Building and refining the Inner Ring that can support the Pentium II C-nodes of the Outer Ring provides a proof of the "fault-tolerance infrastructure" concept.



## 5. EXTENSIONS AND APPLICATIONS

The Inner Ring and D-nodes of DiSTARS offer what may be called a "plug-in" fault-tolerance infrastructure for the client system, that uses contemporary COTS high-performance, but low-coverage processors with their memories and supporting chipsets. The infrastructure is in effect an analog of the human immune system [12] in the context of contemporary hardware platforms [8]. DiSTARS is an illustration of the application of the design paradigm presented in [12].

A desirable advance in processor design is to incorporate an evolved variant of the infrastructure into the processor structure itself. This is becoming feasible as the clock rate and transistor count on chips race upward according to Moore's Law. The external infrastructure concept, however, remains viable and necessary to support chip-level sparing, power switching, and design diversity for hardware, software, and device technologies.

The high reliability and availability that may be attained by using the infrastructure concept in system design is likely to be affordable for most computer systems. There exist, however, challenging missions that can only be justified

1 if their computers have high coverage with respect to tran-  
2 sient and design faults as well as low device failure rates.

3 Two such missions that are still in the concept and  
4 preliminary design phases are the manned mission to Mars [13]  
5 and unmanned interstellar missions [14].

6 The Mars mission is about 1000 days long. The proper  
7 functioning of the spacecraft and therefore the lives of the  
8 astronauts depend on the continuous availability of computer  
9 support, analogous to primary flight control computers in com-  
10 mercial airliners. Device failures and wear-out are not major  
11 threats for a 1000 day mission, but design faults and tran-  
12 sient faults due to cosmic rays and solar flares are to be ex-  
13 pected and their effects need to be tolerated with very high  
14 coverage, i. e. probability of success. It will also be nec-  
15 essary to employ computers to monitor all spacecraft systems  
16 and perform automatic repair actions when needed [9, 15], as  
17 the crew is not likely to have the necessary expertise and  
18 access for manual repairs. Here again computer failure can  
19 have lethal consequences and very high reliability is needed.

20 Another challenging application for a DiSTARS type fault-  
21 tolerant computer is on-board operation in an unmanned space-  
22 craft intended for an interstellar mission. Since such mis-

sions are essentially open-ended, lifetimes of hundreds or even thousands of years are desirable. For example, currently the two Voyager spacecraft (launched in 1977) are in interstellar space, traveling at 3.5 and 3.1 A. U. (astronomical units) per year. One A. U. is  $150 \cdot 10^6$  kilometers, while the nearest star Alpha Centauri is 4.3 light years, or approximately 63,000 A. U. from the sun. Near-interstellar space, however, is being explored, and research in breakthrough propulsion physics is being conducted by NASA [14].

An interesting concept is to create a fault-tolerant relay chain of modest-cost DiSTARS type fault-tolerant spacecraft for the exploration of interstellar space. One spacecraft is launched on the same trajectory every  $n$  years, where  $n$  is chosen to be such that the distance between two successive spacecraft allows reliable communication with two closest neighbors ahead and behind a given spacecraft (Fig. 10). The loss of any one spacecraft does not interrupt the link between the leading spacecraft and Earth, and the chain can be repaired by slowing down all spacecraft ahead of the failed one until the gap is closed.

1 Additional information appears in A. Avižienis, "The hun-  
2 dred year spacecraft", in Proc. of the 1st NASA/DoD Workshop  
3 on Evolvable Hardware, pages 233-39 (July 1999).  
4  
5

## 6 6. KEY TO THE DRAWINGS

7

8 (a) Figs. 1, 2 and 6 — These block diagrams use the  
9 following designators in common.  
10

11 encircled "X": cluster core

12 encircled "M\*" (15 in Fig. 1): M-cluster

13 encircled "M" (unshaded; 201a, 201b and 201c in Fig. 2,  
14 but 601a, 601b and 601c in Fig. 6): M-node (moni-

15 tor-node), powered

16 encircled "M" (shaded): M-node, unpowered (spare)

17 encircled "D" (13 in Fig. 1): D-node

18 encircled "C" (11 in Fig. 1): C-nodes

19 solid black circle with an associated tangential line (14  
20 in Fig. 1): adapter-node (A-node)

21 solid black circle with an associated through-line (18 in  
22 Fig. 1): adapter-port (A-port)

1 large bold circle (16 in Fig. 1; 202 in Fig. 2): M-bus

2 larger, fine circle (17 in Fig. 1; but 203 in Fig. 2):

3 A-lines

4 IP: inner-ring power

5 S in square: power switch

6 S3: set of S3-nodes.

7  
8 Additional item in Fig. 1:

9 12 outer-ring bus

10  
11 Additional items in Fig. 6:

12 602 IC-bus

13 603 disagree lines, internal-error lines, clock lines  
14 and replacement-request lines.

15  
16  
17 (b) Fig. 3 — The following explanations apply to the  
18 n-bit comparator and switch. Section (1) of the drawing is  
19 the symbol only; section (2) shows the detailed structure.

20  
21 c is an n-bit self-checking comparator

22 d is a set of n tristate driver gates

1    if  $\underline{x} = y$ , then  $e = 1$  and  $f = \underline{x}$   
 2    if  $\underline{x} \neq y$  or if  $c$  indicates its own failure,  
 3                                    then  $e = 0$  and  $f = Z$  (high impedance).  
 4  
 5  
 6

7            (c) Fig. 4 — The following explanations apply to both  
 8    of Figs. 4a and 4b.  
 9

10    Internal Blocks:

- 11    401. Encoder
- 12    402. Encoder Register
- 13    403. ID Number for A-pair (ROM)
- 14    404. Comparator (self-checking)
- 15    405. Address Register
- 16    406. Decoder
- 17    407. Command Register
- 18    408. Sequencer
- 19    409. A-line Encoder & Sequencer
- 20    410. Majority Voter

Outputs:

- 441a. Messages to C-(or D-)  
Node via CS 1
- 442a. Node Power On/Off  
Command via CS 2 (C-  
or D-node power)
- 443a. A-node Power Off  
Command to A-pair  
Fuse
- 444a. A-line to M-nodes  
(directly)

|    |                                     |                                 |
|----|-------------------------------------|---------------------------------|
| 1  | 411-414. Input Registers            | 445a. Messages to M-nodes       |
| 2  | 415. Outer Ring Power Switch        | via CS 3 and the                |
| 3  | 416. Inner Ring Power Fuse          | M-bus                           |
| 4  |                                     | 446. Outer Ring Power (to       |
| 5  |                                     | C-node)                         |
| 6  |                                     |                                 |
| 7  | <u>Inputs:</u>                      | <u>Inputs for A-ports Only:</u> |
| 8  | 421a.-424a. From M-bus              | 436a. Error Signal from CS 4    |
| 9  | 425a. Inner Ring Clock              | 437a. Error Signal from CS 5    |
| 10 | 426a. Inner Ring Power              | (these error signals are        |
| 11 | (via Fuse)                          | shown in Figure 9)              |
| 12 | 427a. Power Switch Status           |                                 |
| 13 | 428a. Error Signal from CS 1        |                                 |
| 14 | 429a. Error Signal from CS 2        |                                 |
| 15 | 430a. Error Signal from CS 3        |                                 |
| 16 | 431a. Inputs from C-(or D-) node    |                                 |
| 17 | 432. Disagreement Signal from Voter |                                 |
| 18 | 433. Message from C-(or D-) Node    |                                 |
| 19 | 434. Comparator Output              |                                 |
| 20 | 435. Command to Sequencer           |                                 |
| 21 | 450. Inner Ring Power               |                                 |
| 22 | 451. Outer Ring Power               |                                 |

1       The Clock (425a), Power (426a) and Sequencer (408) out-  
2 puts are connected to all internal blocks. To avoid clutter,  
3 those connections are not shown.

4  
5   Additional note for Figure 4a: Elements 436a, 437a are on the  
6 A-ports only.

7  
8   Additional notes for Figure 4b:

- 9   (1) The A-nodes a and b, and all blocks shown here (except  
10 the C-node), form one ASIC package.
- 11   (2) Inputs 443a or 443b permanently disconnect IR Power from  
12 an A-pair.
- 13   (3) The input and output numbers refer to Fig. 4a.

14  
15  
16   (d) Fig. 5 — Below are explanations for Fig. 5. The  
17 Clock (520), Power (533) and Sequencer (508) are connected to  
18 all Internal Blocks. To avoid clutter, those connections are  
19 not shown.



Internal Blocks:

501. IC-Bus Buffer Storage

502. System Time Register

503. M-Cluster Status Register

504. Outer Ring Status Register

505. ROM Response & Power-up Sequence Store

506. M-bus Buffer Store

507. Input Buffer Store

508. Sequencer (State Machine) and BIST

509. Output Buffer Store

510. A-line Input Buffer Store

511. Power Switch (controlled by  $k$  inputs from S3 nodes) that works on the "summation" principle of three-valued inputs: the three possible values of  $s_i$  ( $i = 1, 2, \dots, k$ ) are ON = +1, OFF = -1, tristate = 0.

The Switch is ON when  $\sum_{i=1}^k s_i \geq +1$ .

Inputs:

520. Clock from S3 nodes

521. Power Switch Control from S3 nodes ( $k$  nodes)

522. M-Bus ( $n$  lines)

- 1 523.-524. A-lines from first A-pair
- 2 525.-526. A-lines from Nth A-pair (the total number of pairs
- 3 of A-lines is N)
- 4 527. "Disagree" signals from other M-nodes (4)
- 5 528. Internal or BIST error signals from other M-nodes
- 6 529. "Start BIST" command from S3 nodes
- 7 530. "Resume TMR" (or Duplex, or Simplex) commands from S3
- 8 531. "Power-Up Outer Ring" command from S3
- 9 532. IC-Bus (j lines)
- 10 533. Inner Ring Power (from switch)
- 11
- 12 Outputs:
- 13 540. to IC-Bus (j lines)
- 14 541. to M-Bus (n lines)
- 15 542. Power Switch Status to S3 nodes
- 16 543. Replacement Request to S3 nodes
- 17 544. Internal or BIST error to other M-nodes and S3 nodes
- 18 545. "Disagree" signal to other M-nodes and S3 nodes

1           (e) Fig. 7 — The following explanations apply to Fig. 7  
2   only. Outputs 721 through 730 are connected in a wired-"OR"  
3   for all four S3 nodes.

4  
5   Internal Blocks:

6   701. Fault-Tolerant Clock (one for both a and b sides),  
7       connected to all Internal Blocks (connections not shown)

8   702a. System Time Counter

9   703a. Interval Timer (for power-off intervals)

10   704a. Outer Ring Status Register

11   705a. M-Cluster Status Register

12   706a. Sequencer (State Machine) with outputs to all Internal  
13       Blocks (connections not shown)

14   707. Backup Power Source, common for a and b sides (connected  
15       to all Internal Blocks, connections not shown)

16  
17   Inputs:

18   710. Clock signals from 3 other S3 nodes

19   711. From IC-Bus (j lines)

20   712. Power Switch Status from M-nodes (5)

21   713. Internal or BIST error signals from M-nodes (5)

22   714. "Disagree" signals from M-nodes (5)

1 715. Replacement Request Signals from M-nodes (5)  
2 716. Power-Off signal from critical event sensors (excessive  
3 radiation, power instability, etc.) or from system  
4 operator  
5 717. Power-On signal (same sources as 716)  
6 718. Primary Inner Ring Power (connected to all Internal  
7 Blocks, connections not shown)

8  
9 Outputs:

10 720. Clock signal to 3 other S3 nodes (connected to all  
11 Internal Blocks, connections not shown)  
12 721. System Time to IC-Bus  
13 722. Interval Time to IC-Bus  
14 723. Outer Ring Status to IC-Bus  
15 724. M-Cluster Status to IC-Bus  
16 725. "Start BIST" Command to M-nodes  
17 726. "Resume TMR" (or Duplex, or Simplex) command to M-nodes  
18 727. "Power Up Outer Ring" command to M-nodes  
19 728. "M-Cluster is Dead" message to system operator  
20 729. Power Switch control for all A-nodes  
21 730. Power Switch control to M-nodes (5 lines)

1  
2 (f) Fig. 8a — At Start, only the S3-nodes are powered  
3 and produce clock signals. There are  $3 + \underline{n}$  unpowered M-nodes,  
4 where  $\underline{n}$  is the number of spare M-nodes originally provided.  
5 Figs. 2 and 6 show  $\underline{n} = 2$ .

6 When the Power On sequence is carried out after a preced-  
7 ing Power Off sequence, then the MC-SR contains a record of  
8 the M-node status at the Power-Off time, and the M-nodes that  
9 were powered then should be tested first.

10  
11  
12 (g) Fig. 8b — The sequence is repeated for all A-pairs  
13 until all C-nodes and D-nodes of the Outer Ring have been tes-  
14 ted and the OR-SR (504) contains a complete record of their  
15 status. The best sequence is to power on and test the D-nodes  
16 first, followed by the top priority (operating system) C-  
17 nodes, then the remaining C-nodes. If the number of powered C-  
18 and D-nodes is limited, the remaining good nodes are powered  
19 off after BIST and recorded as "Spare" in the OR-SR. The OR-SR  
20 contents are also transferred to the S3 nodes at the end of  
21 the sequence.

1           (h) Fig. 8c — This sequence is carried out when the  
2 input 716 is received by the S3 nodes, i.e., when a catas-  
3 trophic event is detected or when the DiSTARS is to be put  
4 into a dormant state with only the S3 nodes in a powered con-  
5 dition, with System Time (702a) and a power-off Interval Timer  
6 (703a) being operated.

7  
8           (i) Fig. 9 — This D-pair replaces the C-node in Figure  
9 4b to show how the A-ports are connected to the D-nodes. The  
10 Twin D-nodes and their A-ports form one ASIC package. The  
11 Outer Ring Power 446 and the Sequencer and Clock 901a are con-  
12 nected to all Internal Blocks.

13  
14 Internal Blocks:

15 901a. Sequencer and Clock

16 902a. Input Buffer Store

17 903a. Encoder of Messages to M-nodes (M-Cluster)

18 904a. Decision Algorithms: Exact and Inexact (N-Version)

19 Comparators and Voters

20 905a. Storage Array for D-node Logs

21 906a. Output Buffer Store

22 907a. Decoder of Messages from M-Cluster

1  
2    Inputs:

3    426 Inner Ring power (via Fuse 416)  
4    441 Messages from A-port to D-node  
5    446 Outer Ring power (from Power Switch 415)  
6    910 Decision Requests and Messages from C-nodes

7  
8    Outputs:

9    431 Messages from D-node to M-nodes (via A-ports)  
10   436 Error Signal from CS4  
11   437 Error Signal from CS5  
12   911 Decision Results and Messages to C-nodes

13  
14  
15  
16  
17       It will be understood that the foregoing disclosure is  
18   intended to be merely exemplary, and not to limit the scope of  
19   the invention -- which is to be determined by reference to the  
20   appended claims.